Tablut Hall of Raids
THoR team

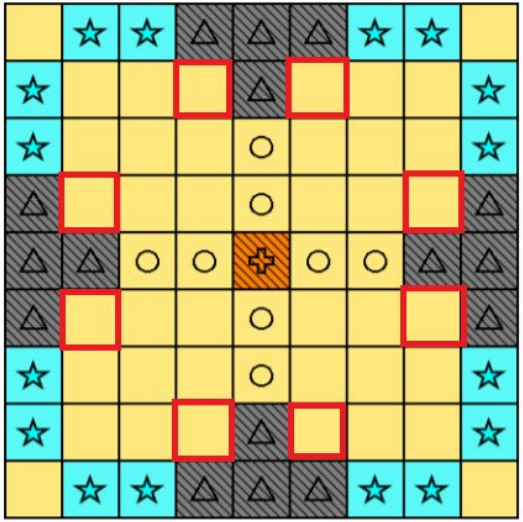
Algorithm and Strategy

General algorithm

- MinMax algorithm with alpha-beta pruning
- Iterative deepening
- Language: started with Python, moved to Java
- BitState as data structure
- Parallelization: first level is explored in parallel:
  - Better usage of resources
  - Deeper levels reached
  - Loss of possible cuts

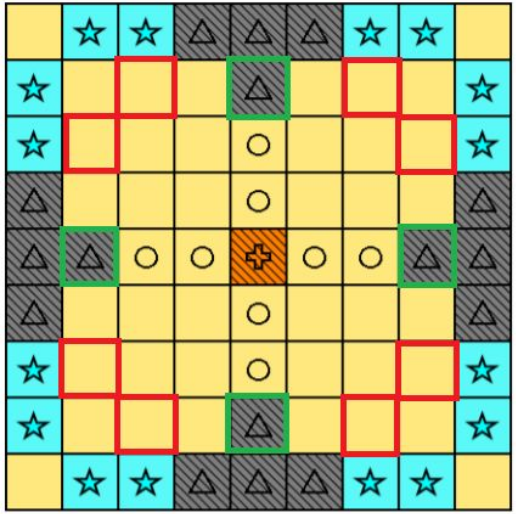
Heuristics (white conditions)

- Remaining Pieces
- “Aggressive King”: king tries to have open paths to exits
- Camp-blocks



Heuristics (black conditions)

- Remaining pieces
- Exits blocks
- “Keep the fronts”: front camps are less likely to be moved



Further ideas for improvement

The following ideas were evaluated, but eventually abandoned for time constraints, unintuitive implementation given the data structures or worse performance.

- Dictionary of state and actions: only kept for avoiding draws
- Lazy SMP approach: use threads to fill dictionary, employ the symmetry.
- King/last moves heatmap: keep into account the king movements
- Iterative blockade: black should keep reducing the space of actions of white